**# Import necessary libraries**

**import numpy as np**

**import pandas as pd**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.linear\_model import LinearRegression, LogisticRegression**

**from sklearn.metrics import mean\_squared\_error, accuracy\_score, classification\_report, confusion\_matrix**

**# Generate example dataset**

**np.random.seed(42)**

**# Assuming 5 features related to health indicators**

**X = np.random.rand(100, 5)**

**# Generating a binary target variable indicating the presence (1) or absence (0) of a disease**

**y = np.random.randint(2, size=100)**

**# Split the data into training and testing sets**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**# Linear Regression**

**linear\_model = LinearRegression()**

**linear\_model.fit(X\_train, y\_train)**

**# Make predictions on the test set**

**y\_pred\_linear = linear\_model.predict(X\_test)**

**# Evaluate the Linear Regression model**

**mse\_linear = mean\_squared\_error(y\_test, y\_pred\_linear)**

**print(f'Linear Regression Mean Squared Error: {mse\_linear}')**

**# Logistic Regression**

**logistic\_model = LogisticRegression()**

**logistic\_model.fit(X\_train, y\_train)**

**# Make predictions on the test set**

**y\_pred\_logistic = logistic\_model.predict(X\_test)**

**# Evaluate the Logistic Regression model**

**accuracy\_logistic = accuracy\_score(y\_test, y\_pred\_logistic)**

**conf\_matrix\_logistic = confusion\_matrix(y\_test, y\_pred\_logistic)**

**classification\_report\_logistic = classification\_report(y\_test, y\_pred\_logistic)**

**print(f'Logistic Regression Accuracy: {accuracy\_logistic}')**

**print(f'Logistic Regression Confusion Matrix:\n{conf\_matrix\_logistic}')**

**print(f'Logistic Regression Classification Report:\n{classification\_report\_logistic}')**